CS448B Critique: Geometric Compression Through Topological Surgery

Peter Chou

February 11, 2000

1 Citation for Paper

Taubin, G. and Rossignac, J. Geometric Compression Through Topological Surgery. *ACM Transactions on Graphics*, Vol. 17, No. 2, April 1998, pp. 84-115.

2 Synopsis

This paper details a geometric compression technique for triangulated 3D polyhedral models which significantly reduces their storage and transmission costs. The method is based on entropy and run-length encoding of vertex and triangle spanning trees. This losslessly compresses mesh connectivity to less than two bits per triangle on average. Vertex positions and properties are quantized and encoded by entropy and predictive coding based on ancestors in the vertex tree.

3 Background

Geometric compression is useful for reducing the cost of storage and network transmission time of 3D polyhedral models. The algorithm in this paper operates on triangle meshes, the prevalent form of representation for polyhedral models. A triangle mesh is defined by three components: vertex positions, vertex connectivity, and photometric properties. Connectivity is encoded losslessly, whereas vertex positions and properties are quantized and thus encoded with (controlled) loss of information.

Mesh compression is related to the problem of mesh simplification, which reduces representation size by reducing the number of vertices and triangles in the model. However, simplification alters mesh connectivity, whereas geometric compression does not. The techniques are actually orthogonal since geometric compression may be applied to simplified versions of a model.

The idea of geometric compression is not new. For example, triangle strips are a basic form of connectivity encoding since they allow previously specified vertices to be reused in defining successive triangles. Quantization of vertex positions and photometric properties was initially proposed by Deering in [2].

The major contribution of Taubin and Rossignac's paper is that it proposes a technique that offers superior compression rates compared to earlier methods.

4 Summary

4.1 Overview of the Technique

The compressed representation of the mesh is composed of vertex and triangle spanning trees, the compressed vertex positions, and the triangle tree marching pattern.

Construction of Spanning Trees There are several heuristics that can be used for constructing the vertex spanning tree. For example, each edge can be assigned a cost equal to edge length or distance to initial vertex, and a standard minimum cost spanning tree construction algorithm can be used. However, for the best compression ratios, it is desirable to minimize the number of branches in the tree. The best results are obtained by "spiraling" out in concentric rings from an initial vertex that forms the root of the tree. The dual graph of the mesh resulting from cutting through the edges of the vertex spanning tree forms the triangle spanning tree.

Vertex Tree Encoding The vertex tree consists of runs of nodes with single children, interconnected by branching and leaf nodes. The tree can thus be efficiently represented by run-length encoding. The tree is traversed in pre-order, and the length of each vertex run is recorded along with two bits to specify branching and leaf nodes.

Compressing Vertex Positions Vertex positions are quantized as done in [2] and stored in the order they are visited in the vertex spanning tree using linear prediction from previous vertices in the tree. The corrective terms are recorded using entropy coding. Photometric properties associated with the vertices are encoded similarly.

Triangle Tree Encoding The triangle tree is encoded in the same way as the vertex spanning tree, but without a branching bit. As the tree is traversed, a marching bit pattern is recorded. Marching bits specify whether the left or right vertex is shared between successive triangles. The marching pattern is compressed by entropy coding.

Decompression and Reconstruction A table of vertex positions is reconstructed from the encoding of the vertex tree. During traversal of the vertex tree, the bounding loop is represented as a table of references to the vertex table. Branch boundary lengths are computed as a preprocessing step. Triangles are then reconstructed from the triangle tree and marching pattern.

4.2 Topological Assumptions

The technique summarized above assumes that the mesh is a triangulated connected oriented manifold without boundary having Euler characteristic 2. It is proved in the paper that with this type of mesh, cutting through the edges of the vertex tree results in a triangulated, simply connected polygon. This is necessary for the above technique to work.

The paper does offer several extensions to handle triangular manifold meshes of Euler characteristic other than 2, nonorientable, and with boundaries. These involve making extra cuts and using some additional bits. Thus, the extended technique is capable of handling meshes with a wide variety of topological characteristics.

4.3 Results

The paper provides a complete set of results, including compression ratio and performance timings. The authors' implementation performs decompression at 60-90K triangles per second. This is reasonable since typical models can be decompressed in less than a second. Compression rates average under 2 bits per triangle and produce typical ratios of 50:1.

5 Comments

The paper is well-written and very thorough. The presented technique is novel, practical, and achieves excellent compression results, making it a useful technique for improving storage or network transmission costs. Furthermore, the algorithms are presented with enough detail to make independent implementation straightforward.

The major weakness of the approach in the paper is that it doesn't offer progressive transmission. This feature is especially desirable in applications involving network transmission. Several other papers covered in the class such as *Progressive Meshes*, *Progressive Simplicial Complexes*, and *Progressive Forest Split Compression* offer alternative representations that allow for progressive transmission at the cost of slightly worse compression rates.

Another weakness of the approach is that decompression can not be performed locally, so the compressed representation can not be streamed and incrementally decompressed. This increases latency since the entire compressed representation must be received before decompression. In addition, by requiring multiple passes and random access to the mesh vertices, memory requirements become larger. Therefore, the technique is unsuitable for cost-effective hardware implementation.

A few minor improvements can be made, which may improve compression rates a little further. First, the run-lengths of the spanning trees may be represented as variable-length tags rather than their explicit value using fixed-length words. For example, if there is a run of length 1024, the approach in the paper would require a 10 bit field to represent the length of each run. If there are many short runs, it would be more efficient to represent them with variable-length words so that fewer bits are necessary to represent commonly occurring short runs.

Another potential area of improvement lies in the quantization of vertex positions and properties. Rather than quantizing uniformly, a variable quantization grid dependent on local feature size can be used to reduce distortion resulting from vertex quantization. This was performed in [1].

6 Discussion Questions

- Traditionally, geometric compression techniques restrict themselves to lossless encoding of connectivity. What are the advantages and disadvantages of doing this?
- Lossy compression of vertex positions results in geometric distortion of the model. How can we determine the point when it would be better from a rate-distortion perspective to simplify the model instead?

References

[1] M. M. Chow. Optimized geometry compression for real-time rendering. In *IEEE Visualization '97*, 1997.

[2] M. Deering. Geometry compression. In *SIGGRAPH 95 Conference Proceedings*, pages 13–20. ACM SIGGRAPH, 1995.